

## REMARKS

Applicants have studied the Office Action dated July 15, 2005. This is an amendment with a Request for Continued Examination (RCE) in response to the August 22, 2007 decision of the Board of Appeals affirming the Examiner. No new matter has been added. It is submitted that the application is in condition for allowance. By virtue of this amendment, claims 3-13, 16-19, and 21-25 are pending. Reconsideration and further examination of the pending claims in view of the above amendments and the following remarks is respectfully requested. In the Office Action, the Examiner:

- Rejected claims 1-3, 14-16, and 20 under 35 U.S.C. §103(a) as being unpatentable over Meyerzon et al. (U.S. Patent No. 6,638,314), in view of Lawrence et al. (U.S. Patent No. 6,289,342) and in further view of Blumenthal (U.S. Patent No. 6,026,409); and
- Rejected claims 4-6 and 17-19 under 35 U.S.C. 103(a) as being unpatentable over Meyerzon et al. (U.S. Patent No. 6,638,314), in view of Lawrence et al. (U.S. Patent No. 6,289,342) as applied to claim 1 and in further view of Hobbs (U.S. Patent No. 6,523,022).

### Rejection under 35 U.S.C. §103(a) in view of Meyerzon, Lawrence, with Blumenthal

As noted above, the Examiner rejected claims 1-3, 14-16, and 20 under 35 U.S.C. §103(a) as being unpatentable over Meyerzon et al. (U.S. Patent No. 6,638,314), in view of Lawrence et al. (U.S. Patent No. 6,289,342) and in further view of Blumenthal (U.S. Patent No. 6,026,409).

Meyerzon discloses a mechanism for obtaining information pertaining to electronic documents that reside on one or more server computers. In particular, Meyerzon teaches that a web crawler program including a gatherer process for gathering information pertaining to electronic documents. See Meyerzon at col. 8, lines 58-60. In the system of Meyerzon, worker threads process URLs and then pass each URL to a filter daemon. See Meyerzon at col. 9, lines 13-16.

The filter daemon uses the URL to retrieve the electronic document at the address specified by the URL. See Meyerzon at col. 9, lines 16-20. After retrieving an electronic document, the filter daemon parses the electronic document and returns a list of text and properties. See Meyerzon at col. 9, lines 29-31. The worker thread then passes the list of properties and text to the indexing engine for creating an index which is used by the search engine in subsequent searches. See Meyerzon at col. 10, lines 13-16. A user may then examine the list of documents returned by the search engine, select a document, and then the web browser displays the selected document to the user. See Meyerzon at col. 8, lines 23-25 and 32-35.

In the present Office Action, the Examiner states on page 2 that Meyerzon teaches

retrieving a web document at an address and extracting contents of the web document for rendering an intermediate dynamically constructed in-memory web page representation of the web document at a hub processing unit which is formatted as if displayed for viewing on an end-user's web browser (Meyerzon Col 7 Lines 60-65 and Col 8 Lines 15-20 i.e. web crawler program searches remote server computers connected to the network for electronic documents and retrieves electronic documents and associated data and a browser displays documents to a user); loading secondary documents associated with the web document in order to render the secondary documents as part of the in-memory web page representation (Meyerzon Col 8 Lines 26-35 i.e. the client computer transmits data to a search engine, the search engine examines its associated index to find documents and returns the documents which are secondary documents and lists the documents for the user to view), wherein the secondary documents include one or more images with textual content embedded therein (Meyerzon Col 9 Lines 44-50 i.e. visual element m include text and hyperlink to an image); analyzing and summarizing the in-memory web page representation to produce a text map for the web page document of the textual contents (Meyerzon Col 10 Lines 13-16 i.e. passes the lists of properties and text to the indexing engine and the indexing engine creates an index, which is used by the search engine in subsequent searches).

The Applicants have cancelled claim 1, which renders the above rejection moot. To further prosecution and respond to the suggestion on page 5 of the August 22, 2007 decision from the Board of Appeals, the Applicants have clarified the "for rendering clause" to positively recite each element necessary to carry out the rendering step i.e. a web crawler with a web browser and

a scripting engine on a server rather than a client to build a dynamic web page prior to indexing the web page. Newly added independent claim21, recites:

*A method for indexing dynamic data documents, the method comprising:*  
*retrieving, to a server, with a web crawler from a network address, a*  
*dynamic data document with client-side scripting code therein;*  
*executing, at the server, a web-browser, as part of the web crawler,*  
*wherein the web-browser displays an in-memory copy of the dynamic data*  
*document which has been retrieved, wherein the in-memory copy of the dynamic*  
*data document maintains a web-browser display format and a web-browser*  
*display layout of the dynamic data document when displayed in the web-browser;*  
*executing, at the server instead of a client system, a browser scripting*  
*engine as part of the web-browser for loading content as directed by the client-*  
*side scripting code into the in-memory copy creating a final web-browser display*  
*representation of the dynamic data document so that the final web-browser*  
*display representation is substantially similar to when the dynamic data document*  
*is viewed by a user in the user's web-browser running on the client system when*  
*all the dynamic data is viewed; and*  
*indexing, at the server, the content in the memory, wherein the content*  
*being indexed is the content which has been loaded by the browser scripting*  
*engine in order to index the dynamic data document as if being viewed by the user*  
*in the user's web-browser on the client system.*

Support for this amendment can be found in the Specification as originally filed at page 4, lines 28-29 to page 5, lines 1-10; page 8, lines 28-29 to page 9, lines 1-2; page 10, lines 1-10 and 12-23; and page 12, lines 8-29 to page 13, lines 1-8 and FIGs. 3 – 8. No new matter has been added.

The Applicants would like to first point the Examiner to the claim element of:

*executing, at the server, a web-browser, as part of the web crawler,*  
*wherein the web-browser displays an in-memory copy of the dynamic data*  
*document which has been retrieved, wherein the in-memory copy of the dynamic*  
*data document maintains a web-browser display format and a web-browser*  
*display layout of the dynamic data document when displayed in the web-browser*

Nowhere does Meyerzon teach executing a web-browser at the server as part of the web crawler. Meyerzon merely discloses a conventional web crawler that searches remote server computers for electronic documents. The web crawler of the present claimed invention, on the other hand, utilizes a web-browser for displaying an in-memory copy of a dynamic data document. Dynamic

data documents comprise executable client side software code that is to be executed by a client's web-browser. See the background of the present invention. A conventional web crawler cannot properly analyze dynamic data documents. In other words, with a conventional web crawler disclosed by Meyerzon, the summarization process fails or produces flawed results when the document itself contains executable client side software code. The reason for this is that the client side software code (e.g., JavaScript, VBScript, or equivalent) is targeted to be executed and interpreted within a web browser's scripting engine. Therefore, the presently claimed invention executes, at the server, a web-browser, as part of the web crawler. Accordingly, the presently claimed invention distinguishes over Meyerzon for at least these reasons.

Furthermore, nowhere does Meyerzon teach or suggest "*wherein the web-browser displays an in-memory copy of the dynamic data document which has been retrieved, wherein the in-memory copy of the dynamic document maintains a web-browser display format and a web-browser display layout of the dynamic data document when displayed in the web-browser*". The Examiner states in the present Office Action on page 2 that Meyerzon teaches that:

[a] web crawler program searches remote server computers connected to the network for electronic documents and retrieves electronic documents and associated data and a browser displays documents to a user".

The Examiner further states in the Examiner's Answer dated November 21, 2006 that

As the appellant points out, rendering "is not implying a visual display of a document, but rather the construction of a data structure of the webpage in memory, which is subsequently analyzed and summarized (Appeal Brief: page 17, first full paragraph)." Although it is unclear precisely how the document is rendered as if displayed for viewing on an end user's web browser merely by construction a data structure of the webpage in memory, it is irrelevant. As the appellant admits, Meyerzon discloses a gatherer, extractor, and summarizer at a hub processing unit (Appeal Brief: page 12, diagram). The diagram fails to show rendering at the hub processing unit.

However, Meyerzon discloses retrieving a document and a filter daemon parsing the electronic document (column 9, lines 29-31). First, retrieving a document to the hub, would construct a data structure of the webpage in memory,

as simply storing a document in memory generates the data structure holding the document. However, upon parsing the retrieved electronic document, a list of text and properties is generated (column 9, lines 29-31). the text is the text that is to be displayed in a web browser program and metadata describing the formatting of the text for display in the browser (column 9, lines 33-35).

The Applicants respectfully suggest that the Examiner is mischaracterizing Meyerzon in view of the presently claimed invention and is also mischaracterizing the present invention. The statement of “Although it is unclear precisely how the document is rendered as if displayed for viewing on an end user’s web browser merely by construction a data structure of the webpage in memory, it is irrelevant.” clearly supports this misunderstanding.

In fact, how the dynamic data document is represented in memory is very important. The presently claimed invention is not merely “construct[ing] a data structure of the webpage in memory, as simply storing a document in memory generates the data structure holding the document” as the Examiner states Meyerzon is doing. The dynamic data document is copied into the memory of the server as is represented the same way as is if the dynamic data document is being displayed on a screen. This ability enhances existing document gathering and analysis by, for example, dramatically improving the quality of the extracted metadata. This is due to the fact that the summarization of a document is based on the whole and complete document as it was designed by the document’s author; the static heterogeneous data as well as the problematic dynamic data is completely rendered and integrated into the metadata for subsequent indexing of all metadata by a web crawler. For example, a dynamic in-memory representation of the web page, as intended to be seen by an end user, is created to extract the most accurate and comprehensive data set possible. A standard web crawler, as taught by Meyerzon, is not able to compose this type of highly dynamic and distributed document that includes dynamic information such as client side script, applets, or their equivalents.

Meyerzon merely teaches that a web crawler retrieves electronic documents and a filter daemon parses the electronic documents for returning a list of text and properties. Meyerzon explicitly

states that text and properties are obtained from tags within the HTML documents. See Meyerzon at column 9, lines 9-43. Therefore, Meyerzon is working on HTML source code, as compared to an *“in-memory copy of the dynamic data document which has been retrieved”*. The information is passed to an indexing engine which creates an index of the retrieved data. The index contains reference information and pointers to corresponding electronic documents, for example, keywords. See Meyerzon at col. 8, lines 1-16. Therefore, Meyerzon clearly does not teach an *“in-memory copy of the dynamic data document which has been retrieved”* as recited for the presently claimed invention. Accordingly, the presently claimed invention distinguishes over Meyerzon for at least these reasons as well.

Even further, Meyerzon does not teach or suggest *“...wherein the in-memory copy of the dynamic document maintains a web-browser display format and a web-browser display layout of the dynamic data document when displayed in the web-browser...”* As discussed above, Meyerzon is merely working on HTML code, which is the different than a representation of the code itself on a browser *“maintain[ing] a web-browser display format and a web-browser display layout of the dynamic data document when displayed in the web-browser...”* Stated differently, Meyerzon is working on the source HTML code, where as the present invention is working on content after the source code is displayed by a web-browser. The two results are often different, especially in the case of dynamic web content. For example, look at the source code of a web page from a dynamic e-commerce site such as www.ebay.com. The static HTML code retrieved from eBay is not the final code used to display the page because of scripting and other dynamic content is loaded only when the web page is displayed in a web browser. Accordingly, the presently claimed invention distinguishes over Meyerzon for at least these reasons as well.

The Applicants would now like to point the Examiner to the claim element of:

*executing, at the server instead of a client system, a browser scripting engine as part of the web-browser for loading content as directed by the client-side scripting code into the in-memory copy creating a final web-browser display*

*representation of the dynamic data document so that the final web-browser display representation is substantially similar to when the dynamic data document is viewed by a user in the user's web-browser running on the client system when all the dynamic data is viewed; and*

*indexing, at the server, the content in the memory, wherein the content being indexed is the content which has been loaded by the browser scripting engine in order to index the dynamic data document as if being viewed by the user in the user's web-browser on the client system.*

Meyerzon is completely silent on this claim element. In fact, Meyerzon is incapable of performing this claim element because the web crawler of Meyerzon does not include a browser scripting engine or functionalities of a web browser. Any client-side scripting is performed on the client side of Meyerzon. The presently claimed invention, on the other hand, executes a browser scripting engine to load content into the memory as directed by the client-side scripting code. For example, if a webpage requires Java code to be executed at the client in order to display additional data to the user, the browser scripting engine (which is part of the web-browser of the web crawler) at the server executes this Java code in order to include the additional data in the in-memory copy of the webpage.

Stated differently, the execution of the browser scripting engine allows content to be loaded into memory as directed by the client-side scripting code to create a final web-browser display representation of the dynamic data document. This final web-browser display representation is substantially similar to when the dynamic data document is viewed by a user in the user's web-browser running on the client system when all the dynamic data is viewed. Meyerzon merely parses the HTML components of a document to create a list of text and properties. See Meyerzon at col. 9, lines 30-59. Accordingly, the presently claimed invention distinguishes over Meyerzon for at least these reasons as well.

The Applicants would also like to point out that the Examiner's statement on page 3 of the present Office Action that Meyerzon teaches that:

the client computer transmits data to a search engine, the search engine examines its

associated index to find documents and returns the documents which are secondary documents and lists the documents for the user to view), wherein the secondary documents include one or more images with textual content embedded therein (Meyerzon Col 9 Lines 44-50 i.e. visual element m include text and hyperlink to an image

This is not the same as “...loading content as directed by the client-side scripting code into the in-memory copy...” The Examiner used the above statement to argue that Meyerzon teaches secondary documents as recited in claim 1, which has been cancelled. With respect to the presently recited language, the content is loaded into memory as a directed by the client-side code which is being executed by a browser scripting engine at the server. The content is loaded into memory to create a final web-browser display representation of the dynamic data document so that the final web-browser display representation is substantially similar to when the dynamic data document is viewed by a user in the user’s web-browser running on the client system when all the dynamic data is viewed. This allows the server to “inde[x], at the server, the content in the memory, wherein the content being indexed is the content which has been loaded by the browser scripting engine in order to index the dynamic data document as if being viewed by the user in the user’s web-browser on the client system”. Meyerzon merely analyzes its index to identify a list of documents that satisfy a user’s request. These documents are then sent as a list to the user. Accordingly, the presently claimed invention distinguishes over Meyerzon for at least these reasons as well.

Lawrence merely teaches an Autonomous Citation Index (ACI) that autonomously creates a citation index from literature in electronic format. The ACI system autonomously locates new articles, extract citations, identifies citations to the same article which occur in different formats, and identifies the context of citations in the body of articles.

Blumenthal merely teaches a system and method for the visual search and retrieval of digital information within a single document of multiple documents. A viewing window has a first pane that provides a global view of digitally stored information and a second pane that provides a local view of the information. A user submits queries and the keywords entered are displayed on



the user's screen in a particular document as being highlighted.

Hobbs merely teaches a method of dynamically augmenting the contents of file of information on a first network resource. Multimedia information, such as video, audio, graphics and text residing on a plurality of Data Warehouses, RDMS, or ODBA systems is selected. This multimedia information is then linked across the Internet, or other network, to any phrase, work, sentence, numbers, movies, audio, and the like contained in documents on an Internet or intranet web site so that any viewer of a web site can directly access updated information in the Data Warehouse.

Nowhere does Lawrence, Blumenthal, or Hobbs teach or suggest:

*A method for indexing dynamic data documents, the method comprising:  
retrieving, to a server, with a web crawler from a network address, a  
dynamic data document with client-side scripting code therein;  
executing, at the server, a web-browser, as part of the web crawler,  
wherein the web-browser displays an in-memory copy of the dynamic data  
document which has been retrieved, wherein the in-memory copy of the dynamic  
data document maintains a web-browser display format and a web-browser  
display layout of the dynamic data document when displayed in the web-browser;  
executing, at the server instead of a client system, a browser scripting  
engine as part of the web-browser for loading content as directed by the client-  
side scripting code into the in-memory copy creating a final web-browser display  
representation of the dynamic data document so that the final web-browser  
display representation is substantially similar to when the dynamic data document  
is viewed by a user in the user's web-browser running on the client system when  
all the dynamic data is viewed; and  
indexing, at the server, the content in the memory, wherein the content  
being indexed is the content which has been loaded by the browser scripting  
engine in order to index the dynamic data document as if being viewed by the user  
in the user's web-browser on the client system.*

Accordingly, the presently claimed invention distinguishes over Meyerzon alone and/or in combination with one or more of Lawrence, Blumenthal, and Hobbs.

Claims 14 and 20 recited similar to claim 1 and have been cancelled as well. New claims 23 and 25 have been added and recite similar to new claim 21. Therefore, the remarks and arguments above made with respect to new claim 21 are also applicable to the allowance of new claims 23 and 25 and will not be repeated.

For the foregoing reasons, independent claims 21, 23, and 25 distinguish over Meyerzon taken alone and/or in view of Lawrence and/or in view of Blumenthal and/or in view of Hobbs. Claims 3-13, 16-19, 22, and 24 depend from claims 21 and 2, respectively, either directly or by way of an intervening claim. Since dependent claims contain all the limitations of the independent claims, claims 3-13, 16-19, 22, and 24 distinguish over Meyerzon taken alone

and/or in view of Lawrence and/or in view of Blumenthal and/or in view of Hobbs, as well, and the Examiner's rejection should be withdrawn, which withdrawal is respectfully requested.

## **CONCLUSIONS**

In light of the Office Action, Applicants believe these amendments serve a useful clarification purpose, and are desirable for clarification purposes, independent of patentability. Accordingly, Applicants respectfully submit that the claim amendments do not limit the range of any permissible equivalents.

Applicants acknowledge the continuing duty of candor and good faith to the disclosure of information known to be material to the examination of this application. In accordance with 37 CFR §§ 1.56, all such information is dutifully made of record. The foreseeable equivalents of any territory surrendered by amendment is limited to the territory taught by the information of record. No other territory afforded by the doctrine of equivalents is knowingly surrendered and everything else is unforeseeable at the time of this amendment by the Applicants and their attorneys.

Applicants respectfully submit that all of the grounds for rejection stated in the Examiner's Office Action have been overcome, and that all claims in the application are allowable. No new matter has been added. It is believed that the application is now in condition for allowance, which allowance is respectfully requested.

The Commissioner is hereby authorized to change any fees that may be required or credit any overpayment to Deposit Account 09-0441. In view of the preceding discussion, it is submitted that the claims are in condition for allowance. Reconsideration and re-examination is requested.

**PLEASE**, if for any reason the Examiner finds the application other than in condition for allowance, the Examiner is invited to call the undersigned attorney at (561) 989-9811 should the Examiner believe a telephone interview would advance the prosecution of the application.

Respectfully submitted,

Date: October 22, 2007

By: / Jon Gibbons/  
Jon Gibbons  
Registration No. 37,333  
Attorney for Applicants

FLEIT, KAIN, GIBBONS,  
GUTMAN, BONGINI & BIANCO P.L.  
One Boca Commerce Center, Suite 111  
551 Northwest 77th Street  
Boca Raton, FL 33487  
Tel. (561) 989-9811  
Fax (561) 989-9812